Identifying Computer-Translated Paragraphs using Coherence Features

Identifying Computer-Translated Paragraphs using Coherence Features

Hoang-Quoc Nguyen-Son[1,2], Ngoc-Dung T. Tieu[3], Huy H. Nguyen[3], Junichi Yamagishi[1,3,4], and Isao Echizen[1,3]

[1]National Institute of Informatics, Tokyo, Japan

[2]KDDI Research Inc., Saitama, Japan

[3]The Graduate University for Advanced Studies, Kanagawa, Japan

[4]The University of Edinburgh, Edinburgh, United Kingdom

{nshquoc,nhhuy,dungtieu,jyamagis,iechizen}@nii.ac.jp

Abstract

We have developed a method for extracting the coherence features from a paragraph by matching similar words in its sentences. We conducted an experiment with a parallel German corpus containing 2000 human-created and 2000 machine-translated paragraphs. The result showed that our method achieved the best performance (accuracy = 72.3%, equal error rate = 29.8%) when it is compared with previous methods on various computer-generated text including translation and paper generation (best accuracy = 67.9%, equal error rate = 32.0%). Experiments on Dutch, another rich resource language, and a low resource one (Japanese) attained similar performances. It demonstrated the efficiency of the coherence features at distinguishing computer-translated from human-created paragraphs on diverse languages.

1 Introduction

Computer-translated text plays an essential role in modern life. Such artificial text helps people, who use different languages, can communicate each other. Machine-translated systems thus significantly support or even completely relieve human translators and interpreters from time-consuming burden.

Thanks to deep learning, neural machine translation have been drastically progressed recently. However, we can still tell "this text must be automatically translated by a computer and reads strangely" sometimes. The unexpected quality leads readers to confuse or misunderstand the meaning of artificial text comparing with the original meaning such as machine-translated web pages, especially in low resource languages. Enhanced methods are thus needed to identify machine-translated text.

Research on computer-translated text detection has been of interest to the natural language processing community. Most detection methods are aimed at the sentence level and use a tree parser (Chae and Nenkova, 2009; Li et al., 2015) to estimate the naturalness of a text passage. However, the scope of these methods is limited to individual sentences and ignores the relationships among sentences. In contrast, some methods are aimed at identifying the translation by the POS N-gram model (Arase and Zhou, 2013; Aharoni et al., 2014), but they extract features from only a few adjacent words. Other methods identify the translated documents (Nguyen-Son et al., 2017) or generated papers (Labbé and Labbé, 2013) using word distribution. However, such methods are only suitable for huge text.

Of the various levels of text, i.e., word, phrase, sentence, paragraph and document, the paragraph is one of the most important. For instance, paragraphs help readers to quickly receive important content amidst large pieces of text on various topics, such as abstracts of scholarly papers and news summaries. Moreover, the paragraph is the main part of current digital text (e.g., email and online product descriptions). The paragraph level provides more information compared with the sentence level. A large document is often separated into individual paragraphs whose sentences have meanings in common and coherence. In this paper, we are interesting in investigating what differentiate of the coherence from machine-translated and human-written paragraphs.

Our hypothesis is that coherence at the paragraph level is one of the factors causing such "artifacts" of the computer-generated sentences. In order to empirically prove this, we built several classifiers using the coherence features and compare them with ones using previously suggested features.

A machine-translated text has almost the same meaning and structure as a genuine original one, but the use of words is different. Figure 1 illustrates the difference by italic words. Translated artificial text tends to be shorter than the original text, as mentioned by Volansky et al. (2013). A human-created paragraph[1] (p_H) thus tends to have more supplemental words (the words in underlined) than a parallel computer-translated one (p_C). The translated text also uses different similar words (bold). These differences may be the cause of the lower coherence of computer-translated paragraphs.

In this paper, we present a method for detecting computer-translated text at the paragraph level. Our contributions are threefold.

- We propose a metric, $POSMat$, to match similar words for a partial paragraph related to a part of speech (POS) pair.
- We present a matching penalty metric, $MatPen$, to reduce the effect of unmatched words. The $MatPen$ is integrated with the $POSMat$ into a paragraph coherence metric, $ParaCoh$, to estimate the coherence of a paragraph.
- We suggest a method to use the $ParaCoh$ for determining whether a paragraph is translated by a computer.

We evaluated the proposed method on 2000 human-transcribed TED paragraphs[2] in English. We also collected the corresponding 2000 German human-created paragraphs. This text was then translated into English by Google to create parallel machine-generated paragraphs. The method-based $ParaCoh$ surpasses previous methods, which identify not only computer-translated but also paper-generated text. Furthermore, experiments on another rich language (Dutch) produced similar results while a low resource Japanese language achieved even higher performances. They demonstrate the capability of the proposed method for recognizing machine-translated text and for evaluating the quality of machine translators.

[1] https://www.ted.com/talks/anant_agarwal_why_massively_open_online_courses_still_matter/transcript

[2] https://www.ted.com

2 Related Work

Computer-translated text detection task has been interested by numerous researchers. The previous methods are summarized by a taxonomy derived from text granularity falling into sentence, short text, and document.

2.1 Sentence

Most primary methods detect computer-generated text at the sentence level on the basis of a parsing tree. For example, Chae and Nenkova (2009) claimed that human-written sentences have a simpler structure than computer-generated ones. The genuine sentence thus contains shorter main phrases including nouns, verbs, adjectives, adverbs, and prepositional phrases. Therefore, they extracted complexity features from the parsing tree, such as parsing depth and average phrase length, in order to distinguish computer-translated from human-written text.

Li et al. (2015) also used parsing features to identify artificial translated text. They proved that human parsing is more balanced in its structure than machine parsing. They extracted balanced-based features from parsing trees such as the ratio of right nodes to left nodes. The limitation of parsing-based methods is that they generate parsings only for individual sentences. A general tree for multiple sentences (e.g., paragraph, document) cannot be generated. Therefore, they cannot quantify relationships among sentences.

2.2 Short text

Arase and Zhou (2013) suggested a method to estimate the text fluency of computer-translated text. They claimed that translation leads to a "salad" phenomenon (Lopez, 2008). The phenomenon points out the generated text has the same meaning as the original one, but the words in the translation are

Human-created paragraph p_H	"The third ***idea*** *that we* ***have*** is ***instant*** feedback. With ***instant*** feedback, the computer ***grades*** exercises. *I mean*, how else do you teach 150,000 students? ***Your*** computer *is* ***grading*** all *the* ***exercises***. *And* we've all ***submitted*** homeworks, and *your grades come back* two weeks *later, you've* forgotten *all* about it…"
Computer-translated paragraph p_C	"The third ***concept*** is ***called immediate*** feedback. With ***immediate*** feedback, the computer ***rates*** *the* exercises. How else do you teach 150 000 students? ***The*** computer ***evaluates*** all ***tasks***. We've all ***done*** homework and forgotten about it *during the* two-week *correction period. ...*"

Figure 1: Coherence of parallel human-created vs. computer-translated paragraphs. The difference is presented in italic. The using of various similar words is highlighted in bold. The missing words in computer-generated text are described by underline.

more chaotic, so it affects the text fluency. The authors used a POS N-gram language model to quantify the fluency of consecutive words. Nguyen-Son and Echizen (2017) also used a word N-gram model to extract features combining with special features, which they called as noise such as misspelled words. However, such noise often contains in informal conversations. Furthermore, Aharoni et al. (2014) extended the POS N-gram model by integrate function words features for improving computer-translated identification. They argued that automatic translations contain more function words than human-written translations. These N-gram models only extracted features from a limited number (up to three in common) of adjacent words and ignored the coherence between words separated by one or more other words.

2.3 Document

Nguyen-Son et al. (2017) proposed a method to detect a general document using Zipfian law. Word distribution is aligned with Zipfian distribution to distinguish computer- with human-generated text. The authors proved that human-generated text has more adoption with Zipf's law than machine-generated one.

On the other hand, Labbé and Labbé (2013) detected another kind of computer-produced document, i.e., paper generation. They pointed out that generated papers contain many duplicated patterns. They thus estimated word distribution of a candidate paper with both fake and genuine papers on the basis of inter-textual similarity. The nearest similarity was used to determine whether a human or a computer creates the input paper.

The restriction of document-based methods is that they need numerous number of words in order to estimate the word distribution. Their performances are thus decreased in smaller text.

3 Proposed method

The four steps of the proposed method are illustrated in Figure 2.

- **Step 1 (*Separate sentences*)**: The sentences in an input paragraph p are separated using the Stanford CoreNLP splitter (Manning et al., 2014).
- **Step 2 (*Match similar words*)**: Each word is matched with another word in the other sentences if they are associated with a POS pair.
- **Step 3 (*Calculate coherence metric*)**: The similarities of the matched words in each sentence pair are used to calculate the paragraph coherence metric, $ParaCoh$.
- **Step 4 (*Classify the text*)**: $ParaCoh$ is used to determine whether the input p is human-created or computer-translated paragraph.

3.1 Separating Sentences (Step 1)

The sentences s_i in a paragraph p are separated from the paragraph and placed in set S. Separation is done using the Stanford CoreNLP splitter (Manning et al., 2014):

$$S = Split(p) = \{s_i\}. \quad (1)$$

For example, the first two sentences (s_{H_1} and s_{H_2}) in the human-created text example in Figure 1 are separated from the paragraph:

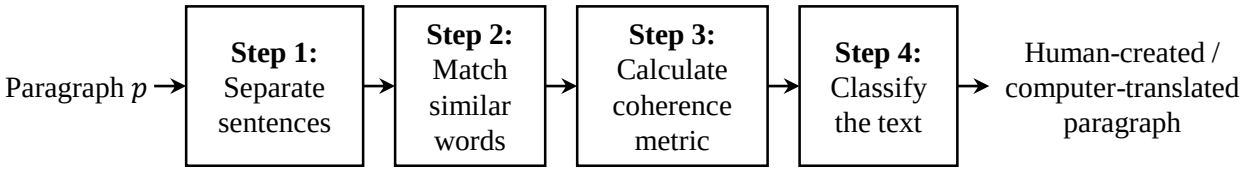

Figure 2: Schema for identifying a computer-translated paragraph.

s_{H_1}: "*The third idea that we have is instant feedback*."

s_{H_2}: "*With instant feedback, the computer grades exercises*."

3.2 Matching Similar Words (Step 2)

English words exist in various grammatical forms which often express similar meaning. For example, a verb "*be*" can be represented by different variants (e.g., "*is*, " "*were*," "*being*," "*'s*"). We use the Stanford lemma tool (Manning et al., 2014) to normalize these words. For example, lemmas of several words are shown from the sentence s_{H_2} and s_{H_4} in Figure 3.

A lemma in a sentence is kept if there is another lemma in another sentence and their POSs conform with a processing POS pair while the other lemmas are removed. For example, suppose the processing pair consists of two plural nouns {NNS, NNS}, and the processing sentences are s_{H_2} and s_{H_4}. Because both sentences each contain a plural noun, these plural nouns are preserved. The other lemmas are eliminated by strikethrough (Figure 3).

A remaining lemma of a sentence is matched with at most one lemma in the other sentence by using the Hungarian algorithm (Kuhn, 1955) to ensure that the contributions of the remaining lemmas are balanced. The algorithm is also used to match the pairs with maximum similarity in total. The similarity of two lemmas is estimated using $path$ metric (Pedersen et al., 2004). This metric is calculated by the shortest path of these lemmas in a Wordnet semantic ontology. For example, a plural noun of s_{H_2} is matched with a corresponding plural noun of s_{H_4} and the similarity of these identical lemmas equals 1.0.

On the other hand, although the meaning of computer-generated text is similar to human-written one (Figure 1). The use of other similar words effects to text coherence and thus influences to the matching. For instance, Figure 4 demonstrates the declined matching of two computer-generate sentence s_{C_2} and s_{C_4} due to the use of another word "*tasks*" in the second sentence. Other similar words also result in the matching degradation of other POS pairs, i.e., the use of "*rates*" and "*evaluates*" (in bold).

3.3 Calculating Coherence Metric (Step 3)

The matching words are used to calculate the POS matching metric $POSMat$ for s_i and s_j:

$$POSMat(s_i, s_j) = \frac{\sum_{w_k \in s'_i, w_l \in s'_j} path(w_k, w_l)}{n}, \quad (2)$$

where w_k and w_l are pair-matched words for the two sentences, n is the number of matched pairs, $path(w_k, w_l)$ is the $path$ similarity metric of the two matched words estimated using Wordnet (Pedersen et al., 2004) while s'_i and s'_j are two sets which contain remaining words in s_i and s_j, respectively. For example, the matching metric of s_{H_2} and s_{H_4} is:

$$POSMat(s_{H_2}, s_{H_4}) = \frac{1}{1} = 1. \quad (3)$$

Since the number of words in s'_i often differs from the number in s'_j, we use a penalty matching metric p based on the machine translation METEOR metric (Denkowski and Lavie, 2010) to reduce the difference:

$$p(s_i, s_j) = 0.5 \times (\frac{|UnMat(s'_i) - UnMat(s'_j)|}{max(|s'_i|, |s'_j|)})^3, \quad (4)$$

where $UnMat(s')$ is the number of nonmatching words. The final matching metric, $POSMat$, is then updated:

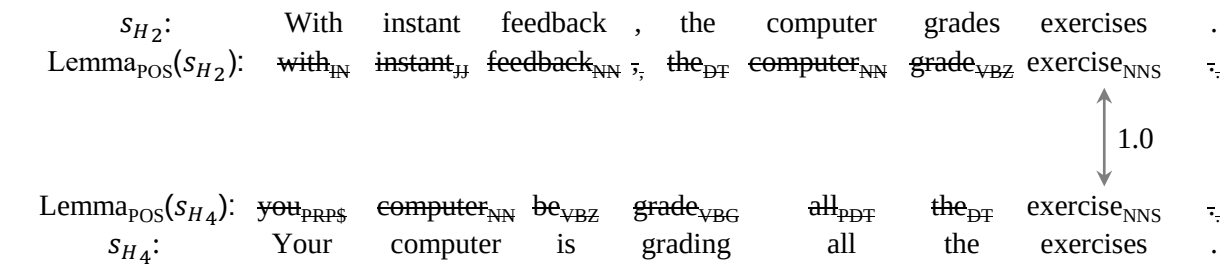

Figure 3: Matching plural nouns in human-generated sentence pair.

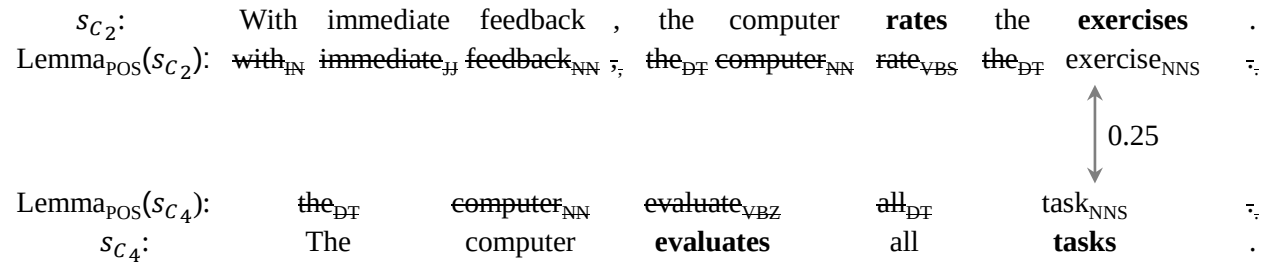

Figure 4: Matching plural nouns in machine-generated sentence pair.

$$POSMat(s_i, s_j) = POSMat(s_i, s_j) \times (1 - p(s_i, s_j)). \quad (5)$$

For example, the matching of s_{H_2} and s_{H_4} is re-estimated using:

$$p(s_{H_2}, s_{H_4}) = 0.5 \times (\frac{|0-0|}{1})^3 = 0, \quad (6)$$

$$POSMat(s_{H_2}, s_{H_4}) = 1 \times (1-0) = 1. \quad (7)$$

Since all candidate words are matched, the p unchanged the similarity matching of s_{H_2} and s_{H_4} related to plural nouns. Figure 5 shows another example of matching adjectives of two human-generated sentences s_{H_1} and s_{H_2}. The $POSMat$ metric of this matching is presented in Equation 9 demonstrating the effect of nonmatching bold word "***third***" into the re-estimated metric.

$$p(s_{H_1}, s_{H_2}) = 0.5 \times (\frac{|1-0|}{2})^3 = 0.125, \quad (8)$$

$$POSMat(s_{H_1}, s_{H_2}) = 1 \times (1 - 0.125) = 0.875. \quad (9)$$

The paragraph coherence metric for a paragraph related to a POS pair is then calculated:

$$ParaCoh(p) = \frac{\sum_{s_i \in p, s_j \in p, s_i \neq s_j} POSMat(s_i, s_j)}{max(1, \binom{m}{2})}) = \frac{\sum_{s_i \in p, s_j \in p, s_i \neq s_j} POSMat(s_i, s_j)}{max(1, \frac{m(m-1)}{2})}, \quad (10)$$

where m is the number of sentences, the denominator presents for the number possible distinguished sentence pairs in paragraph p, and the function max covers a paragraph having only one sentence.

3.4 Classifying the text (Step 4)

The coherence features presented by $ParaCoh$ for each POS combination are used to classify human-created and computer-translated text. Here, large linear classification (LINEAR) (Fan et al., 2008) outperforms other popular classification algorithms. We thus chose LINEAR as the final classifier to determine whether the input paragraph is written by a human or translated by a machine.

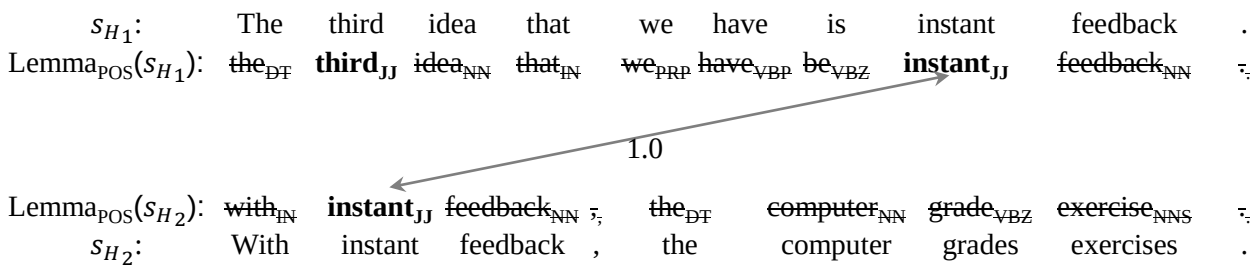

Figure 5: Matching similar nouns in human-generated sentence pair.

4 Evaluation

4.1 Datasets

We created a dataset from 2100 scripts of recent (2013 to 2018) TED talks[3]. These human-created texts were manually transcribed by native English speakers, and then, 2000 paragraphs were randomly extracted. The paragraphs contained 14.41 sentences on average. Then, we collected corresponding 2000 paragraphs translated by native German speakers. These paragraphs had the same content as the English ones. The German paragraphs were then translated into English by Google Translate to create machine-translated paragraphs.

4.2 Comparison with previous methods

Accuracy (ACC) and equal error rate (EER) was chosen as evaluation metrics, since the corresponding F-measures would give equivalent results with the accuracy. Four commonly classifiers, which are mentioned in previous methods, including logistic regression (LOGISTIC), support vector machine optimized by stochastic gradient descent (SGD(SVM), SVM optimized by sequential minimal optimization SVM(SMO), LINEAR were run with 10-fold cross validation. The performances of previous methods and the proposed method are shown in Table 1.

These all methods tended to focus on recognize translation text in different granularity. The first method, which was previously used for document-level, compared word distribution with Zipfian one (Nguyen-Son et al., 2017). In the second method, Li et al. (2015) identified computer-translated text using balancing properties of a parsing tree. Because a parsing tree is only created for individual sentences, the average of these sentence features was used for the paragraph. For larger text, another method quantified the frequency of text by using the POS N-gram model combining with function words (Aharoni et al., 2014).

As shown in Table 1, the document-based method is degraded their performances on lower granularity because the alignment between word frequency and Zipfician distribution is more effective in huge number of words. On the other hand, the sentence-level method using parsing trees were ineffective at the paragraph level as it does not take into account the relationship among sentences in a paragraph. The extension of N-gram features by combining them with function words worked better than the two previous methods, but it only estimates the coherence of consecutive words. In contrast, the proposed method overcomes the problem and achieves the best performance across all classifiers.

The experiments of previous methods are given the identical or competitive results with the chosen classifiers in corresponding methods. These classifiers are thus used for experiments below. With our method, since LINEAR achieved the best performance, it was chosen to create the final classifier.

Table 2 shows the top five performances of 990 in-duplicated POS combination pairs sorted by their accuracies. These pairs affect to the coherence machine-translated German text comparing with human-created text. These pairs should thus be taken more attention in order to improve machine translators in this language.

4.3 Other languages

Finally, we conducted similar experiments with another rich language (Dutch) and lower resource one (Japanese). In each, 2000 human-written and cor-

[3] https://www.ted.com

Method	LOGISTIC		SGD(SVM)		SMO(SVM)		LINEAR	
	ACC	EER	ACC	EER	ACC	EER	ACC	EER
(Nguyen-Son et al., 2017)	64.8%	35.0%	65.0%	**34.8%**	**65.2%**	35.1%	65.0%	34.9%
(Li et al., 2015)	67.7%	**32.5%**	66.2%	34.2%	67.0%	33.6%	**67.8%**	33.4%
(Aharoni et al., 2014)	67.6%	32.3%	66.3%	33.8%	67.4%	32.6%	**67.9%**	**32.0%**
Our	69.7%	30.4%	69.8%	30.6%	70.9%	31.1%	**72.3%**	**29.8%**

Table 1: Comparison with previous methods on accuracy (ACC) and equal error rate (ERR) metrics. The best metrics are shown in bold. The underline describes for the best classifiers, which are chosen in previous methods. The topmost performance is highlighted by red.

Rank	POS pair	ACC	EER
1	VB-VBG	63.7%	39.1%
2	VBG-RB	63.6%	39.5%
3	VBG-NN	63.5%	38.5%
4	VBG-VBG	63.5%	40.9%
5	IN-VBG	62.3%	40.3%

Table 2: Performances of top five POS pairs.

responding machine-translated human-written paragraphs are used. Each paragraph has an average of 14.64 and 14.30 sentences in Dutch and Japanese, respectively. Since the equal error rate metrics are given similar results above, we only show the accuracy metrics in this comparison. Furthermore, we compared with another document-based method (Labbé and Labbé, 2013), which distinguishes another kind of computer-generated paragraph, i.e., paper generation. In contrast with other previous methods, which extracted features and uses classifiers for identifying translated-generated paragraph, Labbé and Labbé compared word distribution of a candidate document with all training distributions using inter-textual distance. The candidate document is labeled as the same type of the nearest comparing. The result of the comparison is shown in Figure 6.

The experimental results showed that the paper generation detector achieves the worst performances. Because the translation generated the text have almost same meaning with the original text, the word distribution is almost synchronized. It significantly affects to the hypothesis of the Labbé and Labbé's method through all three languages. With other methods, two rich resource languages acquired the similar performances. On the other hand, experiments on the lower resource language produced significantly better performances. Our method reached the best results in the three languages. This method can be used to evaluate the quality of translated text in various languages with different resources.

5 Conclusion

The coherence of human-created paragraphs is generally better than that of computer-translated ones. The method we propose quantifies the coherence of sentences in a paragraph by matching similar words. Our evaluation showed that the coherence features result in higher accuracy than that of state-of-the-art methods on different granularity of German text. Moreover, the evaluation of a similar resource language (Dutch) and a low-resource one (Japanese) achieved similar results. It also demonstrated another capability of our method on measuring the quality of machine translators on various languages with different resource-levels. It significantly supports for current translators recognizing and improving the text generation.

To the best of our knowledge, current parallel translation corpora support for sentence-level[4]. Other larger level datasets use English as original text such as TED talk[5]. Our future work targets to create a human translated English dataset from other languages in paragraph level. Then, we use the coherence features to classify human-translated and computer-translated text.

In another direction, the next research includes using a deep learning network to improve the quantification of the coherence and fluency features. We also extend the matching algorithm to phrases for

[4] http://www.statmt.org/europarl/

[5] https://www.ted.com/

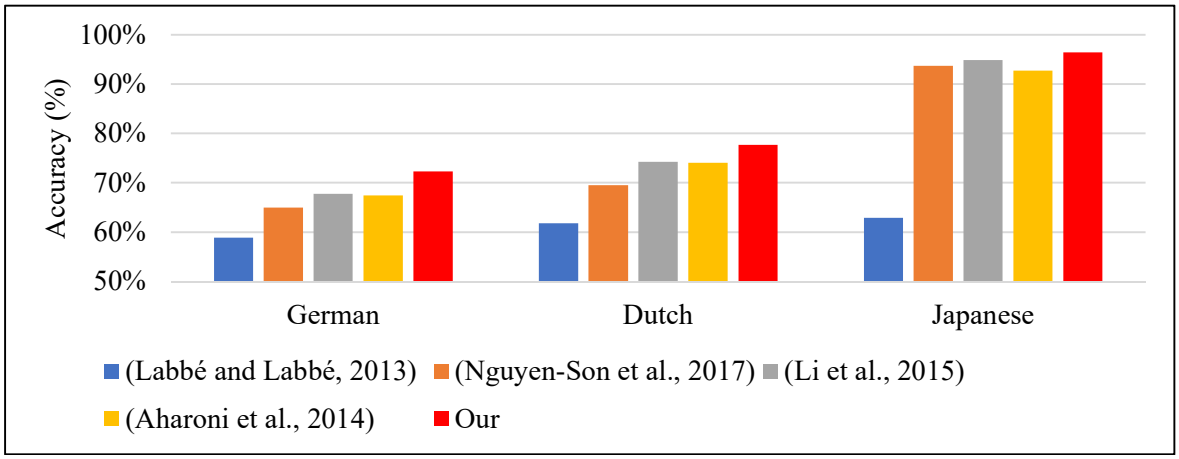

Figure 6: Evaluation on various languages.

further enhancing the coherence metrics.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers JP16H06302 and 18H04120

References

Roee Aharoni, Moshe Koppel, and Yoav Goldberg. 2014. Automatic detection of machine translated text and translation quality estimation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 289–295.

Yuki Arase and Ming Zhou. 2013. Machine translation detection from monolingual web-text. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1597–1607.

Jieun Chae and Ani Nenkova. 2009. Predicting the fluency of text with shallow structural features: case studies of machine translation and human-written text. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 139–147.

Michael Denkowski and Alon Lavie. 2010. Extending the METEOR machine translation evaluation metric to the phrase level. In *Proceedings of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies*, pages 250–253.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9(8):1871–1874.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics*, 2(1-2):83–97.

Cyril Labbé and Dominique Labbé. 2013. Duplicate and fake publications in the scientific literature: how many SCIgen papers in computer science? *Scientometrics*, 94(1):379–396.

Yitong Li, Rui Wang, and Hai Zhao. 2015. A machine learning method to distinguish machine translation from human translation. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pages 354–360.

Adam Lopez. 2008. Statistical machine translation. *ACM Computing Surveys*, 40(3):8.1–8.49.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Hoang-Quoc Nguyen-Son and Isao Echizen. 2017. Detecting computer-generated text using fluency and noise features. In *Proceedings of the 15th International Conference of the Pacific Association for Computational Linguistics*, pages 288–300.

Hoang-Quoc Nguyen-Son, Ngoc-Dung T Tieu, Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. 2017. Identifying computer-generated text using statistical analysis. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, pages 1504–1511.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet:: Similarity: measuring the relatedness of concepts. In *Proceedings of the North American Chapter of the Association for Computa-*

tional Linguistics - Human Language Technologies, pages 38–41.

Vered Volansky, Noam Ordan, and Shuly Wintner. 2013. On the features of translationese. *Literary and Linguistic Computing*, 30(1):98–118.